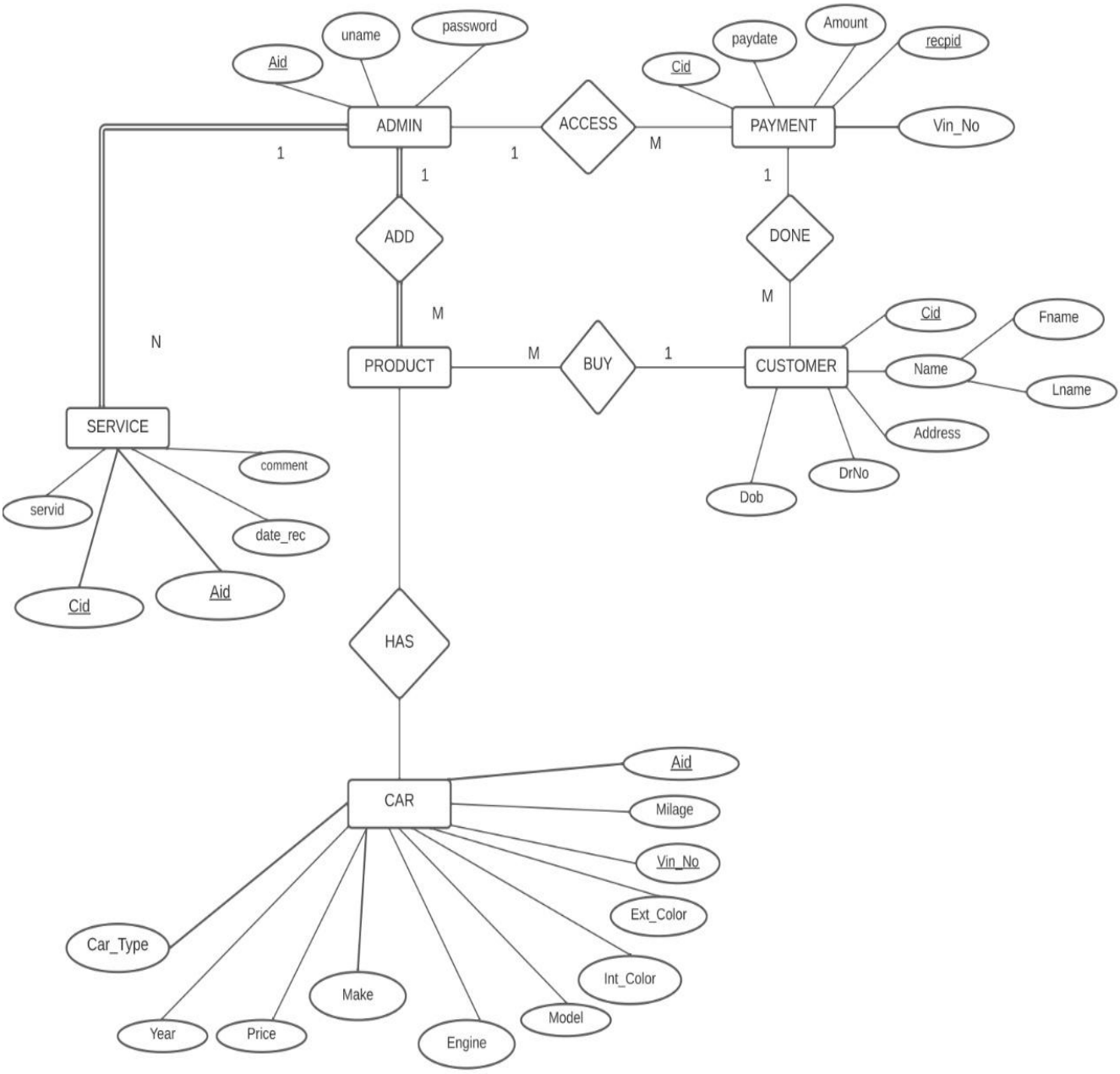
The first step is to understand the requirement of the project and build entity relationship diagram to understand the relationship between the entities.

Requirement:

Build a car inventory of old and new car which also offer car service to its customers and Business rule for the project is that customer age should be greater than 18 then only he can buy the car and price of the car should be greater than zero.

ER Diagram



ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education, and research. It is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system.

Entity: -

It is a definable thing such as person, object, concept, or event. It is often shown as a rectangular shape. Here in our diagram, we have defined 5 entities that are Admin, Service, Car, Payment, and Customer.

Relationships: -

Here we defined how entities are associated with each other. It is often shown as diamond shape.

E.g.: In our diagram we can see the relationship between Admin and Product is admin can add products.

lly, relationship between customer and product is customer can buy the product in the same way relationship between admin and payment is admin can access the payment.

Attribute: -

Attribute is the property/Characteristics of the entity. It is often shown in an oval shape.

In our diagram the Attribute of Admin is Aid, Username, password where Aid is the key attribute (i.e, it is a unique value for each admin).

lly, Attribute for Service is service id (i.e, Servid), date-rec, comments where servid is the key attribute.

lly, Attribute for Customer is Customer id (i.e, Cid) Customer name (Cname), Address, City, State, driver license number (DrNo), date of birth(dob), where customer name is a composite Attribute.

Cardinality: -

Defines a numerical attribute of the relationship between 2 entities or entity set. There are 3 main cardinality relationships that are one-to-one, one-to-many, many-to-many.

In our diagram cardinality between entities Admin and payment is one-to-many that is one admin can access number of Payments, but that payment will be accessed by that admin.

lly, cardinality between entities Admin and product is one-to-many that is one admin can access number of Products, but that product will be accessed by that admin. in a same way customer can buy number of products, but that product is only taken by one customer and so on.